CLAIMS

1. Method of converting virtually concatenated data streams into continuously concatenated data streams, wherein the data is transmitted in containers which are inserted into pulse frames, a sequence of N containers is combined to form a multiframe, each container is provided with a multiframe indicator relating to its position with respect to time within the multiframe, and the virtually concatenated data streams consist of X partial data streams/channels,
characterised in that
containers which are allocated in each case to the same point in the multiframe are identified by evaluation of the multiframe indicator, the time shift of these identified individual containers of the partial data streams with respect to each other is measured, in the event of a shift being present only leading containers are delayed in each case by periods of time which ensure that all containers are aligned with respect to time, and in each channel fill levels of buffer memories are compared with threshold values and in dependence thereon channel-individual stuff-indications are generated and stuffing operations are performed under consideration of the stuff-indications of all channels.

2. Method according to claim 1, characterised in that the time shift of the individual containers is measured using the pointer values of the containers.

3. Method according to any one of the claims 1 to 2, characterised in that the containers are buffered, wherein the writing-in procedure is performed individually for each partial data stream and the reading-out procedure is performed in a synchronized manner for all partial data streams.

4. Method according to any one of the claims 1 to 3, characterised in that the synchronization of the partial data streams is performed by exchanging time reference values and/or stuff-indications and defect indicators.

5. Method according to any one of the claims 1 to 4, characterised in that the communication between the containers as required for the purpose of aligning the containers with respect to time is performed in a decoupled manner with respect to time.

6. Method according to any one of the claims 1 to 5, characterised in that the fill levels are determined by forming an average value over an integer multiple of a pulse frame line.

7. Method according to any one of the claims 1 to 5, characterised in that the fill levels are determined at a defined point in time relative to the outgoing and/or received pulse frame.

8. Method according to any one of the claims 1 to 7, characterised in that buffer memory read pointers are reset in each channel individually by the difference between the dedicated pointer value and the pointer value mentioned by a neighboring channel, in order to increase the dedicated channel delay by this value.

9. Method according to any one of the claims 1 to 8, characterised in that from the incoming containers it is also possible to read and evaluate sequence indicators and/or path traces and accordingly a preconnected switching matrix is controlled.

10. Method according to any one of the claims 1 to 9, characterised in that the read multiframe indicators and/or sequence indicators are filtered for bit-errors.

11. Method according to any one of the claims 1 to 10, characterised in that in order to generate the overheads in the outgoing data streams the data streams which are aligned with respect to time are utilized.

12. Method according to any one of the claims 1 to 11, characterised in that the minimum delay of the containers is raised, in order to compensate for the time difference in the transmission of that data between the channels which is required for the purpose of generating the overheads.

13. Method according to any one of the claims 1 to 12, characterised in that where the buffer memory of at least one channel fails to achieve a fixed minimum fill level, the delay of all partial data streams is increased by a positive-stuffing operation.

14. Method according to any one of the claims 1 to 13, characterised in that where a fixed maximum fill level in each of the channels is exceeded, the delay of all partial data streams is reduced by a negative-stuffing operation.

15. Method according to any one of the claims 1 to 14, characterised in that the fill levels of buffer memories are compared with dynamically adaptable threshold values, wherein in the event that a dynamically adaptable threshold is exceeded:

the dynamically adaptable threshold value is incremented when any other channel fails to achieve a further, predeterminable fixed threshold value,

the delay of all partial data streams is reduced by means of a negative-stuffing operation when the fixed threshold value is exceeded by the fill levels of all channels, and

the dynamically adaptable threshold value is decremented upon the dynamically adaptable threshold value not being achieved and at the same time the fixed threshold value being exceeded by the buffer memory fill levels of all channels.

16. Device for the purpose of implementing the method according to any one of the claims 1 to 15, characterised in that
each channel (KA1, KA2, ...) is allocated a pointer interpreter (PI1, PI2), followed by a buffer memory (ES1, ES2) and a pointer generator (PG1, PG2),

the pointer generators are synchronized with respect to each other, and each pointer generator is arranged for controlling the reading-out from the buffer memory associated with its channel,

a channel which is selected as a master channel (KA1) is provided with an overhead inserter (OI1), to which is supplied the output data from overhead extractors (OE1, OE2) which are disposed downstream of the buffer memories (ES1, ES2),

and the buffer memories (ES1, ES2) are arranged for delaying all of the containers or aligning them with respect to time.

17. Device according to claim 16, characterised in that in the slave channels (KA2, KA3) a fill byte inserter (FSI) follows on in each case from the pointer generators (PG2, PG3).

18. Device according to claim 16 or 17, characterised in that in order to align with respect to time the partial data streams in the pointer interpreters (PI1, PI2), multiframe counters (MFZ) are provided which are synchronized by the multiframe indicators of the input data streams in such a manner as to be bit-error-tolerant.